JPL

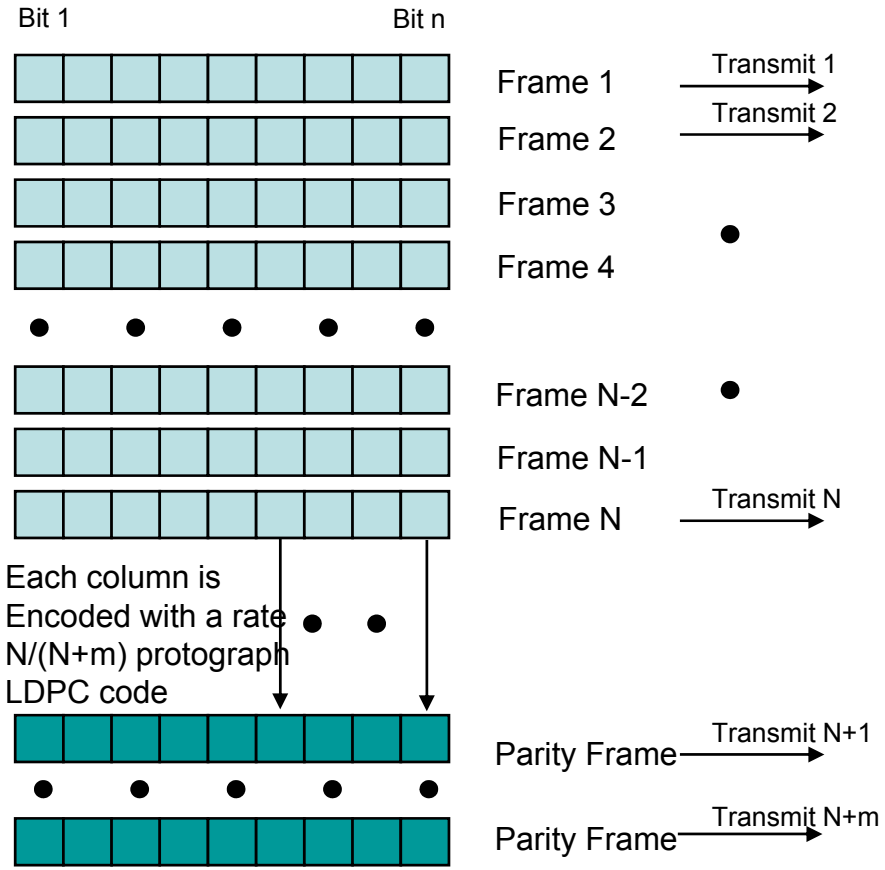
Consultative Committee
CCSDS
for Space Data Systems

Protograph LDPC Codes for the Erasure Channel

Long Erasure Codes BOF

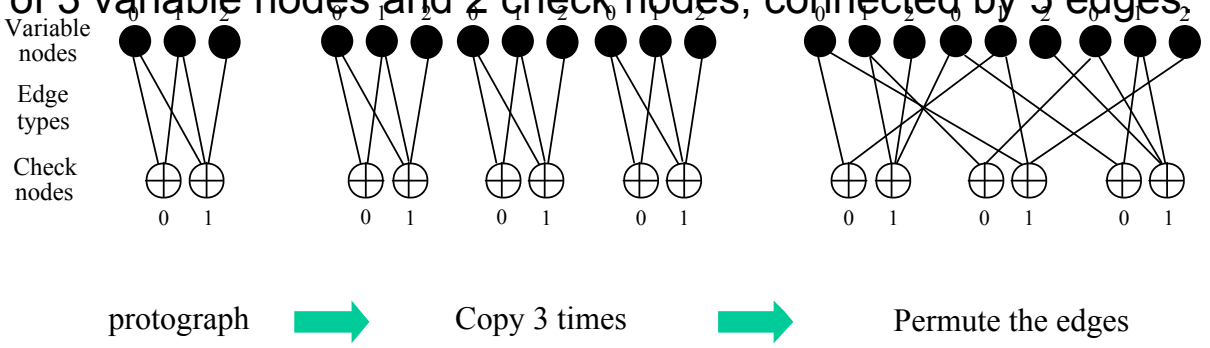
- We first looked at low-rate codes, drawing on our experience in designing such codes for the AWGN channel
- However, high code rates are called for if erasure-correcting codes are used as an outer code to a high-performance inner error-correcting code such as a turbo or LDPC code
- This presentation reviews results (both theoretcal and simulated) for low-rate codes (r ~ 1/2) and introduces new results for high-rate codes (r ~ 9/10 and higher) and for "quantized-rateless" high-rate codes
- Results are valid for both the binary erasure channel (BEC). They are also valid for a block erasure channel (BIEC) or burst erasure channel (BuEC), provided the code size is measured in units of blocks and data blocks are interleaved sufficiently to make block erasures independent throughout a codeword (see next slide for interleaving and encoding method)

- The data frames each of size n are transmitted over a burst erasure channel.
- The encoding method is more effective for channels with random burst erasures of size B < n.
- Assume these bursts occur with probability p.
- The frames are stored as rows of a N x n matrix. Then each column of the matrix is encoded with a (N+m,N) LDPC code.



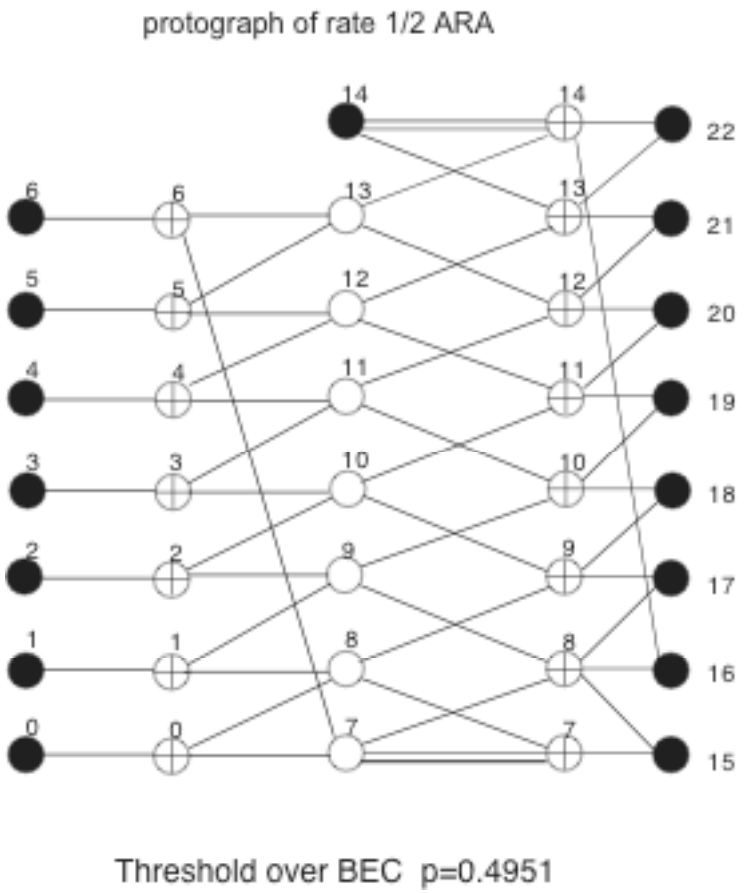
Protograph LDPC codes

• For high-speed decoding, it is advantageous for an LDPC code to be constructed from a protograph. JPL has extensive experience designing protograph LDPC codes for AWGN channels.
• A protograph is a Tanner graph with a relatively small number of nodes. A ``copy-and-permute" operation can be applied to the protograph to obtain larger derived graphs of various sizes.
• This operation consists of first making N copies of the protograph, and then permuting the endpoints of each edge among the N variable and N check nodes connected to the set of N edges copied from the same edge in the protograph.
• The derived graph is the graph of a code N times as large as the code corresponding to the protograph with the same rate and the same distribution of variable and check node degrees.
• As a simple example, we consider the protograph shown in Fig. 1. This graph consists of 3 variable nodes and 2 check nodes, connected by 5 edges.



Low-Rate Protograph Codes

- Protographs can be designed with asymptotic thresholds very close to the capacity limits at low rates (e.g., r = 1/2)
- Rate-1/2 ARA-type protograph at right has asymptotic decoding threshold p* = 0.4951 on the BEC
  - Extremely close to the capacity limit p = 0.5 for any rate-1/2 code on the BEC
- However, simulated performance for reasonable block sizes k = 1000 to 4000 is poor
- A somewhat better choice for finite blocks is a less optimized protograph such as the rate-1/2 AR4JA protograph designed for the AWGN channel
  - asymptotic threshold is poorer, p* = 0.44, but performance is better, with no error floor
  - For AR4JA codes, the minimum stopping set size and the minimum codeword weight both grow linearly with block size



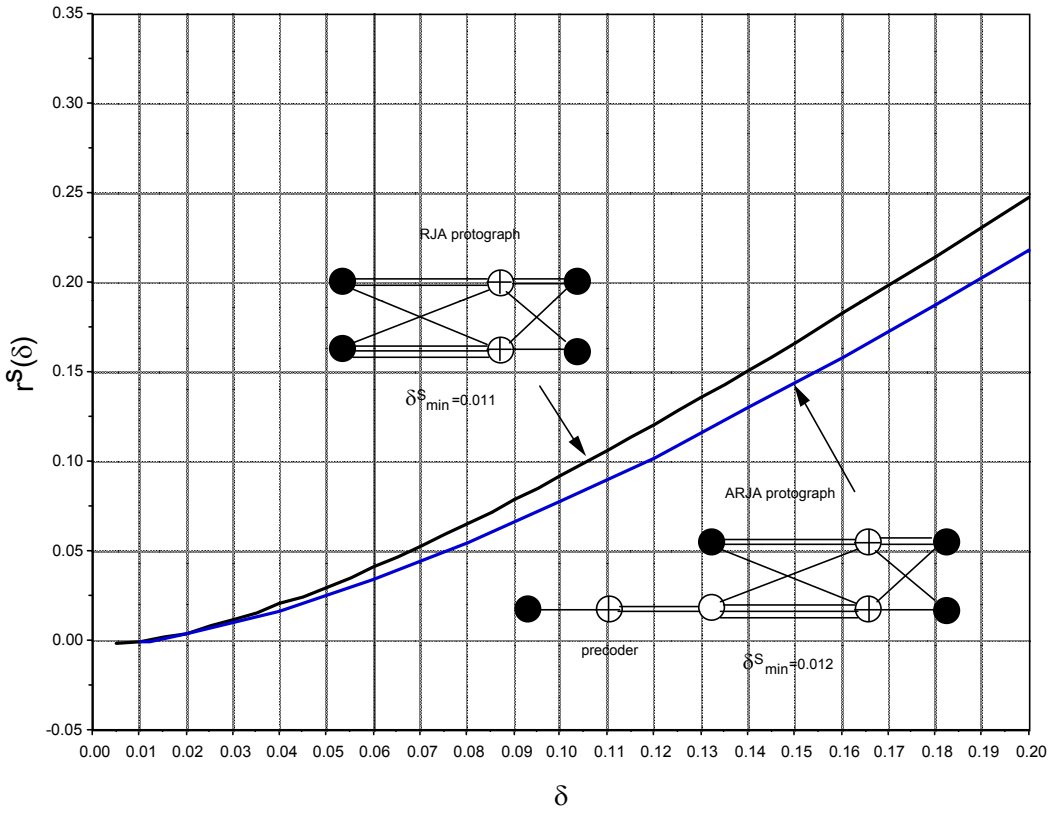
Analysis of Stopping Set Enumerators For Protograph-based Code Ensembles

- Significance of Stopping Sets
  - A set of variable nodes is called a *stopping set* if all its check node neighbors are connected to this set at least twice.
  - Message passing decoding fails whenever all the variable nodes in a stopping set are erased.
  - Thus, for message-passing LDPC decoders, on the BEC channel, the size of a stopping set plays a role similar to that of codeword weights for maximum-likelihood decoders.

- Analysis of ensemble average stopping set size enumerators
  - We aim to design protograph LDPC codes such that the minimum stopping set size of the code grows linearly with block size
    - because codes with linear minimum stopping set have low error floor performance on the BEC when decoded iteratively via message passing
  - To do this, we first need to compute stopping set size *enumerators* for protograph-based LDPC code ensembles
    - Such computational tools have been developed elsewhere for *unstructured* irregular LDPC code ensembles only
  - We developed a new computational method to evaluate stopping set enumerators for structured LDPC codes built from protographs.
  - We used this analysis method to design protograph LDPC codes with linear minimum stopping set size (i.e. minimum stopping set size growing linearly with code block size)

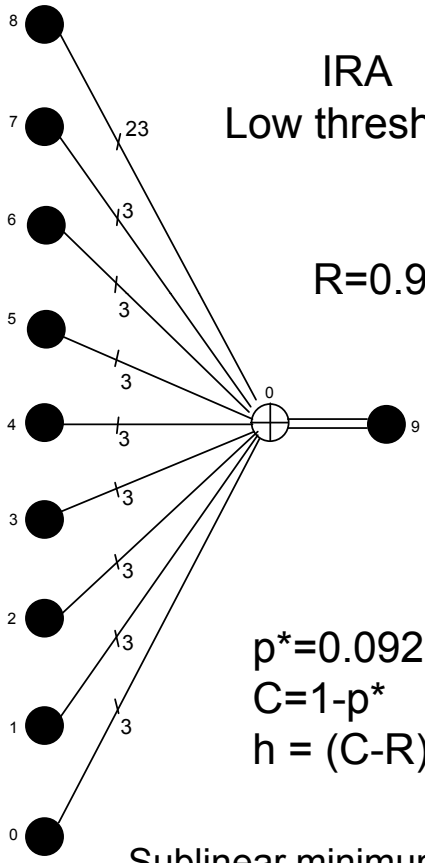
Minimum Stopping Set Size Analysis

- Vertical axis in graph at right is the normalized logarithmic stopping set size distribution of code ensembles built from the give protographs
- Horizontal axis is the normalized stopping set size relative to code length
- If the curve starts negative, the first zero crossing δ_{min} is the coefficient of linear growth of stopping set size with code length
- Many simple protographs (e.g., IRA, ARA) have curves starting with zero or positive slope, hence δ_{min} = 0, i.e., no linear growth of stopping set size with code length
- Both rate-1/2 AR4JA and (slightly less complex R4JA) protographs show linear growth, with coefficients 0.012 and 0.011 respectively.



High-Rate Protograph Codes

- High-rate protographs (e.g., r = 0.9) must be very simple, because there are many variable nodes per check node (e.g., 10:1 ratio for r = 0.9)
  - Many variable nodes in protograph allows little room for expansion by large-sze circulants, unless the overall code size is extremely large
- At high rates, irregular degree distributions
  - give only moderate gains unless they're extremely irregular
  - Give poor performance at finite block sizes if they're extremely irregular
- Example at right is an IRA code (irregular repeat-and-accumulate) with one repeat-23 node and the other 9 repetitions of degree-3, plus an accumulator (degree-2)
  - Asymptotic threshold p* = 0.0927 is close to the capacity limit of p = 0.1 (achieved code rate within h=0.8% of BEC capacity C=1–p* at the asymptotic threshold)
  - By comparison, optimum unstructured LDPC code with degrees 2-100 has p*=0.090, (h=1.08%), but maximum degree 100 (illustrates effectiveness of protograph structure to reduce the required maximum node degree)



IRA
Low threshold

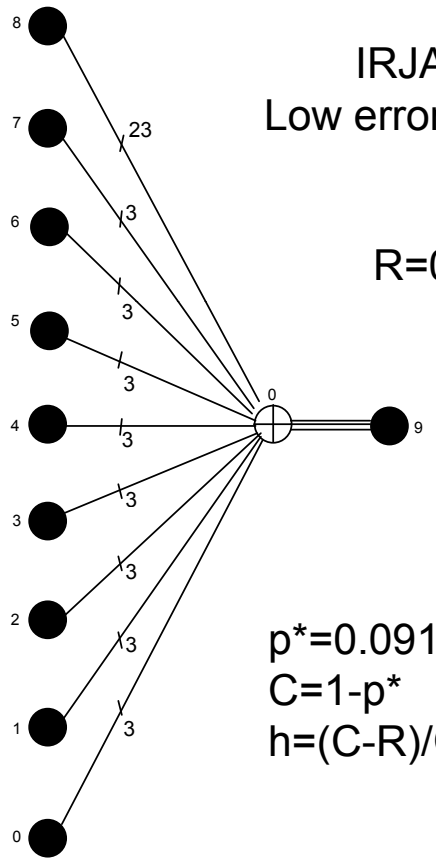
R=0.9

p*=0.0927
C=1-p*
h = (C-R)/C=0.80%

Sublinear minimum stopping set size
(due to degree-2 accumulator)

High-Rate Protograph Codes (cont.)

- Codes designed from IRA-type protographs have sublinear minimum stopping set size
- IRJA-type protograph using a “jagged accumulator” with degree-3 nodes yields linearly growing minimum stopping set size
- IRJA protograph at right has same degree distribution as IRA protograph on previous slide, except for the degree-3 jagged accumulator
  - Asymptotic threshold p* = 0.0919 is also close to the capacity limit of p = 0.1 (achieved code rate within h=0.89% of BEC capacity C=1–p* at the asymptotic threshold)
  - By comparison, optimum unstructured LDPC code with degrees 3-100 has p*=0.083 (h=1.87%), but maximum degree 100. Thus, the IRJA protograph is even more effective at reducing the required maximum node degree compared to unstructured LDPC codes constrained to have no degree-2 nodes (to achieve linearly growing minimum stopping set size)



IRJA
Low error floor

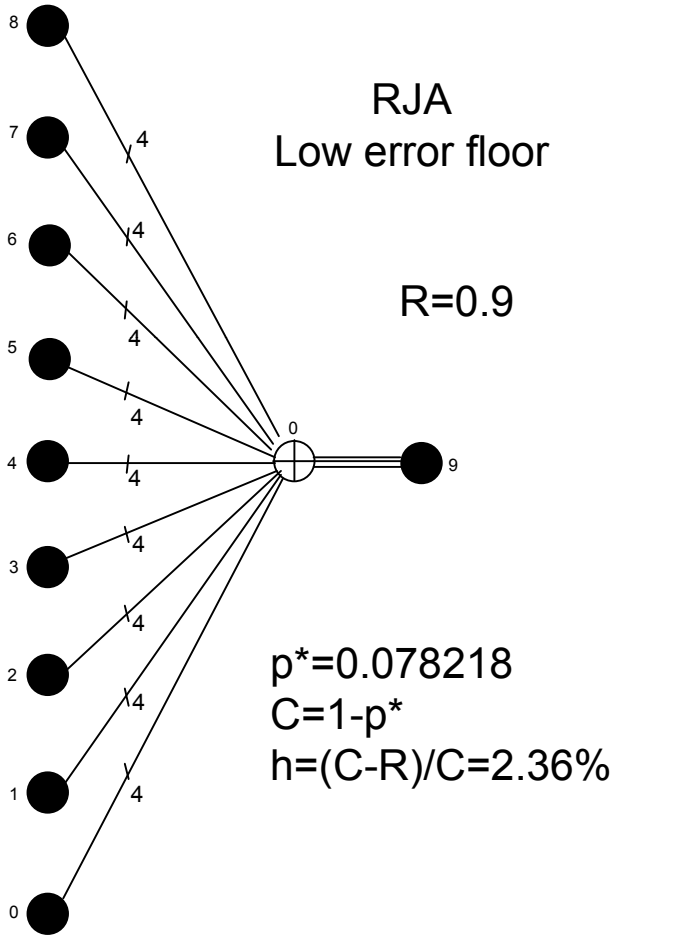
R=0.9

p*=0.0919
C=1-p*
h=(C-R)/C=0.89%

Linear minimum stopping set size

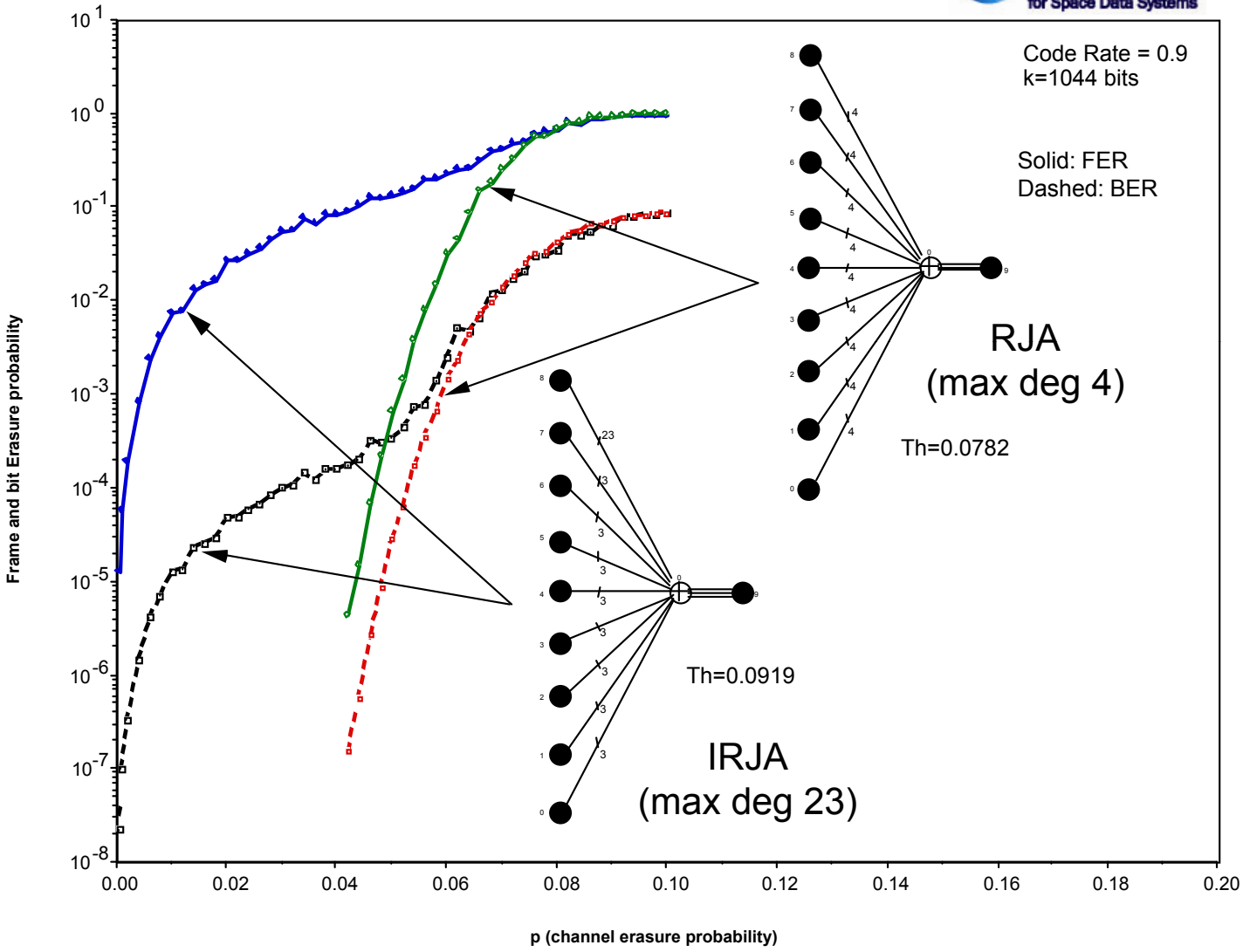
High-Rate Protograph Codes (cont.)

- Despite the good asymptotic thresholds, the degree-23 nodes in the previous protographs cause considerable “expansion suboptimality” when expanded to block sizes of only a few thousand bits (because either the circulants are too small or the graph will have too many short loops)
- The regular RJA-type protograph using all repeat-4 nodes and a “jagged accumulator” with degree-3 nodes is designed to reduce this problem
  - Asymptotic threshold p* = 0.0782 is significantly farther from the capacity limit of p = 0.1 (achieved code rate within h=2.36% of BEC capacity C=1–p* at the asymptotic threshold)
  - This protograph also yields linearly growing minimum stopping set size
  - Performance of codes expanded to about a thousand bits is superior to that of codes obtained from the protographs with better asymptotic thresholds but higher maximum-degree nodes (see next slide for comparison)

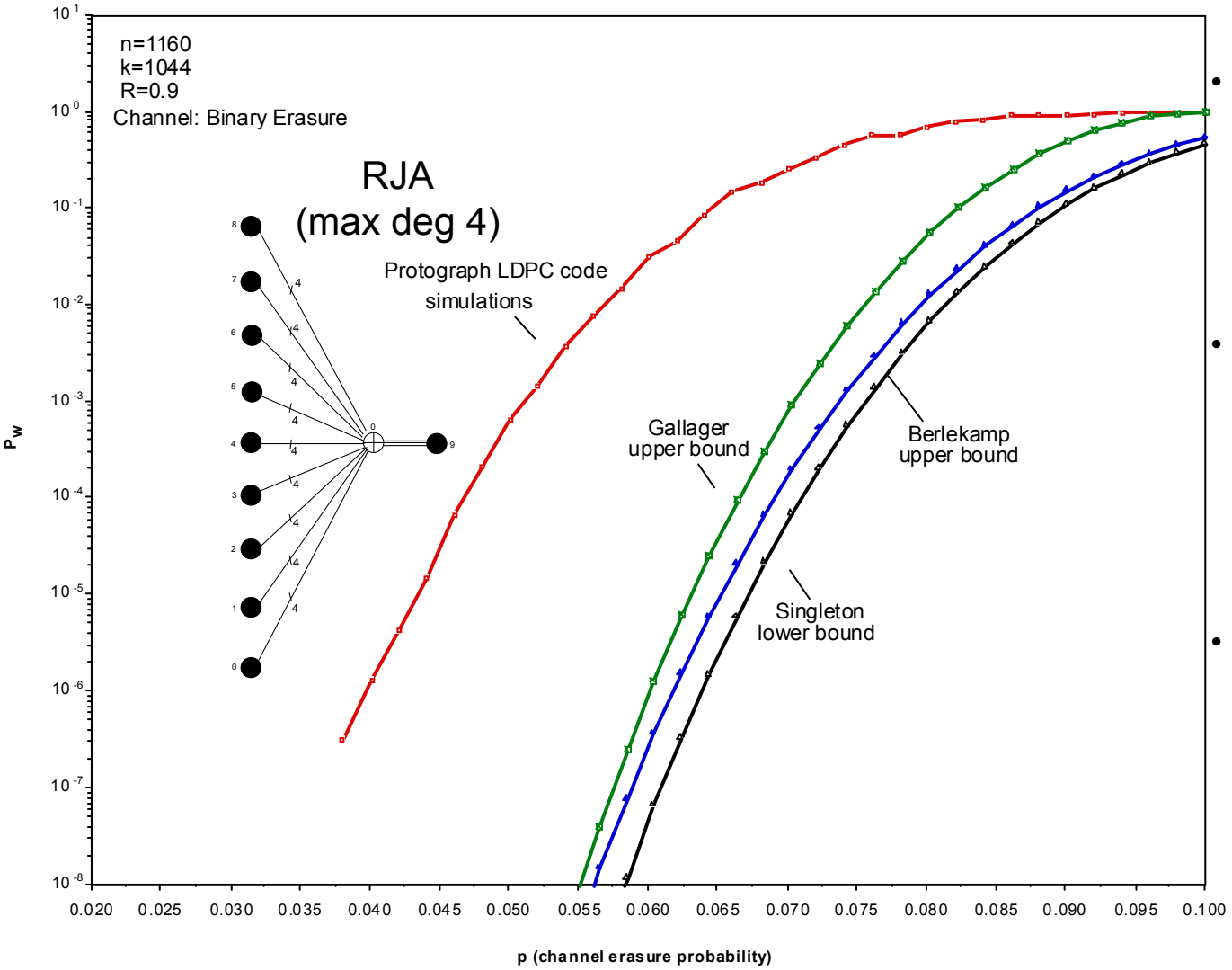


High-Rate Protograph Codes (cont.)

- Both rate-9/10 protographs are expanded to block size k = 1044
- The RJA code (with maximum degree 4) gives acceptable performance
  - still not very close to the asymptotic threshold, due to the relatively small block size
  - compare to bounds for finite block size on next slide
- The IRJA code (with maximum degree 23) gives horrible performance
  - due to many loops in the expanded graph, despite its superior asymptotic threshold



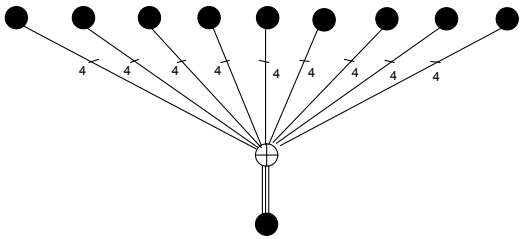
Simulation Results Compared with Bounds for the BEC



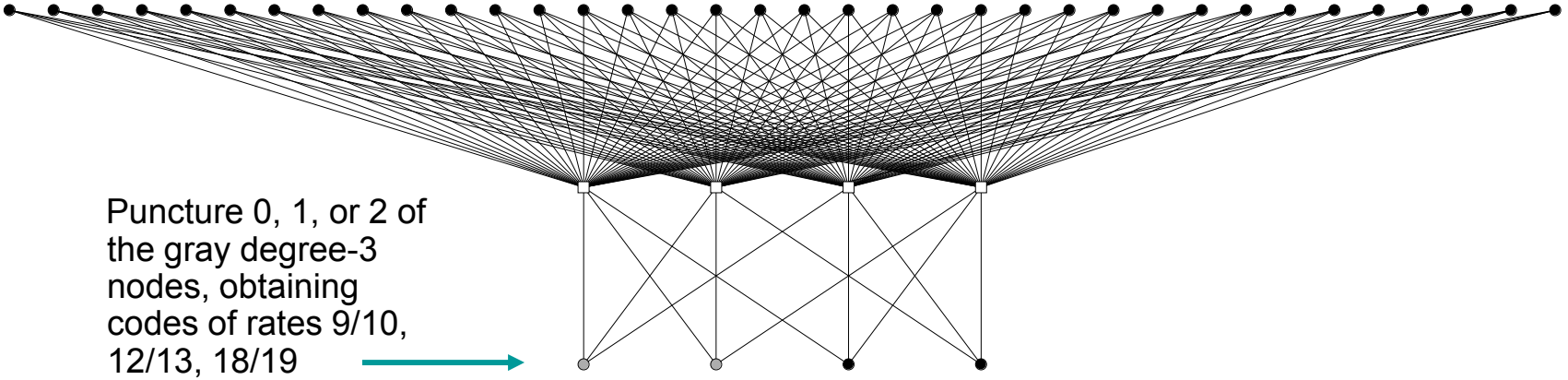
- Upper and lower bounds are for finite-size codes with same rate r = 9/10 and block size k = 1044 as the RJA code tested
- Upper bounds are for average performance of random codes on the BEC. Lower bound limits the performance on the BEC of any code with the given r and k
- There is still a significant difference between the RJA code's performance and the performance predicted by the bounds

Quantized-Rateless High-Rate Codes

- The regular RJA protograph can be punctured to form a “quantized-rateless” code achieving three different discrete rates over a range from 0.9 to ~0.95
  - The three codes are progressive parity codes (like the rateless codes of Digital Fountain). In other words: (1) the same k input bits are encoded by each code; and (2) the coded bits of each higher-rate code are a subset of the coded bits of the next lower-rate code.
  - This gives three different performance options, allowing the code to adapt somewhat to an unpredictable channel (see performance curves on next slide)
  - Unlike fountain codes, this “quantized-rateless” code includes only a discrete set of rates covering a limited rate range



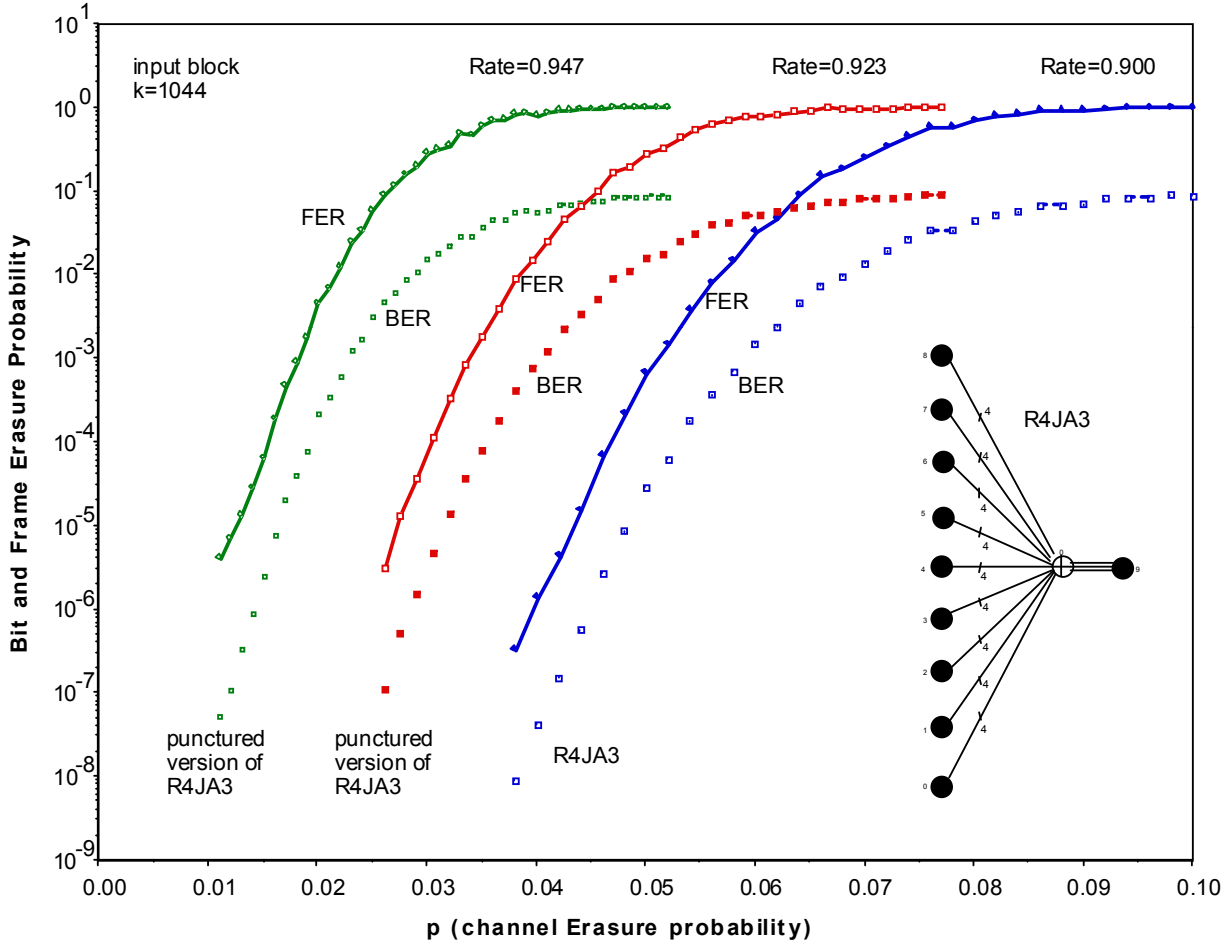
Expand first by 4; then apply circulant permutations to edges of expanded protograph



Puncture 0, 1, or 2 of the gray degree-3 nodes, obtaining codes of rates 9/10, 12/13, 18/19

Performance of Quantized-Rateless RJA Codes

- All three codes were tested with input block size k = 1044
- The quantized-rateless family allows one to seamlessly and efficiently achieve FER of about 10^{-6} with channel erasure probabilities varying from about 0.01 to 0.04



Efficiencies of Rateless and Fixed-Rate Codes on the Binary Erasure Channel

A rateless code communicates k (fixed) information symbols by transmitting n (variable) coded symbols.
LT codes and Raptor codes are examples of rateless codes that are simple to encode and decode.

The inventors of LT and Raptor codes, Luby and Shokrollahi, define the overhead ε of a rateless code on a pure erasure channel as

$$\varepsilon = (k^+ - k)/k$$

where k^+ is the number of coded symbols that are successfully received (not erased)

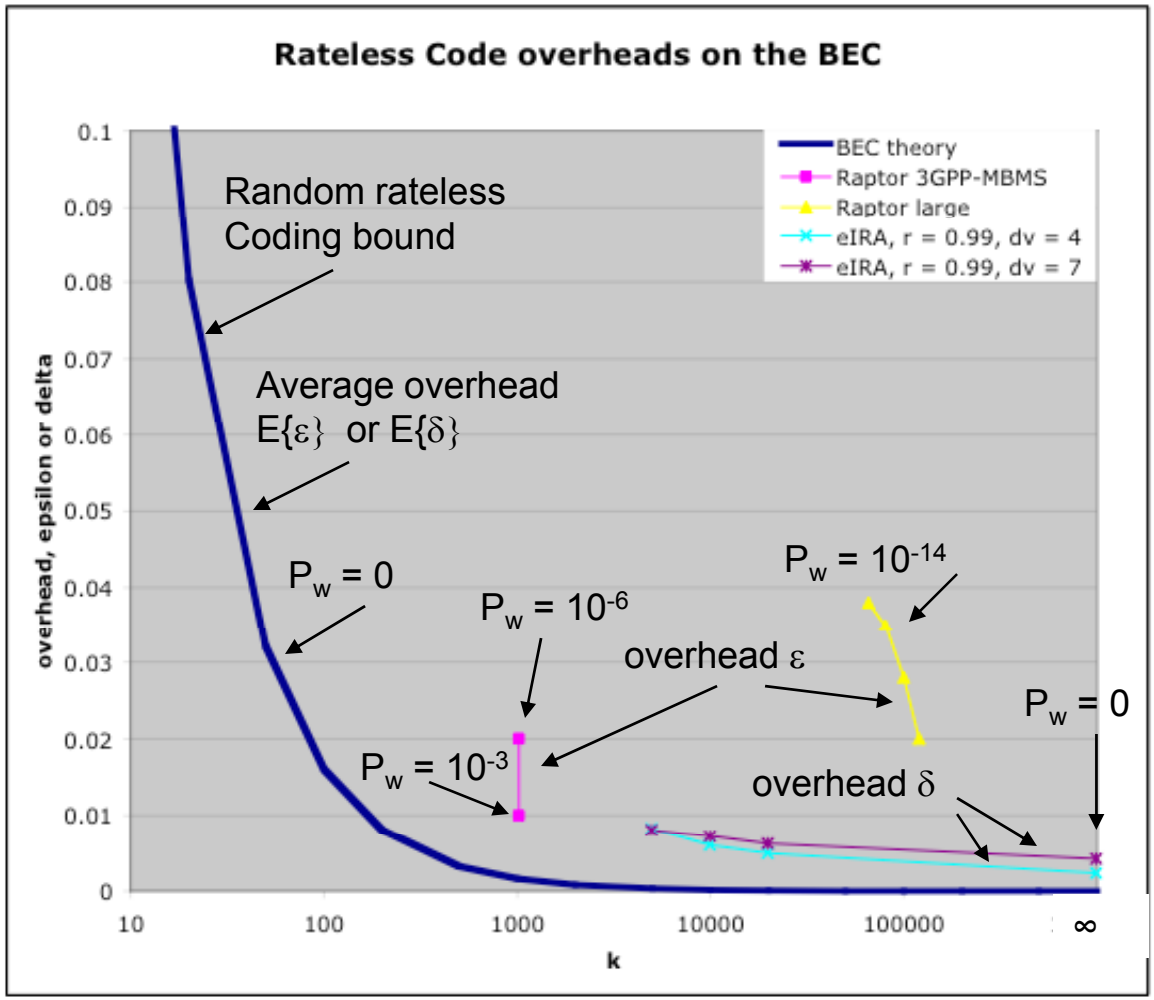
Another measure of overhead is the fraction of excess information at the capacity limit in the transmitted coded symbols

$$\delta = (Cn - k)/k = C/r - 1$$

where C is the capacity of the channel (per transmitted coded symbol), and r is the code rate

Random rateless codes can achieve zero frame erasure probability on the erasure channel with fantastically small average overhead $E\{\varepsilon\} = E\{\delta\}$, even with small finite input blocks, if they are decoded optimally (see graph).

- **LT codes and Raptor codes of unbounded input block size can achieve arbitrarily small frame erasure probability with arbitrarily small overheads $\varepsilon \to 0$ as $k \to \infty$**
- **Large LT codes and Raptor codes with practical finite input blocks can achieve extremely small frame erasure probabilities with maximum overheads of a few percent (see graph)**
- **Smaller Raptor codes (e.g., designed for 3GPP-MBMS standard) can achieve moderately low frame erasure rates with maximum overheads of a few percent (see graph)**
- **Fixed-rate codes of high rate (e.g., the r = 0.99 eIRA codes of Chiani presented at Athens) can achieve low overheads δ, but with only moderately low residual erasure probabilities**


Rateless Code overheads on the BEC
BEC theory
Raptor 3GPP-MBMS
Raptor large
eIRA, r = 0.99, dv = 4
eIRA, r = 0.99, dv = 7
Random rateless
Coding bound
Average overhead
E{ε} or E{δ}
Pw = 0
Pw = 10-6
Pw = 10-14
overhead ε
Pw = 0
Pw = 10-3
overhead δ
overhead, epsilon or delta
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
10
100
1000
10000
100000
∞
k

Conclusions

- For very high code rates and short block sizes, a low asymptotic threshold criterion is not the best approach to designing LDPC codes.
- Simple protographs with much regularity and low maximum node degrees appear to be the best choices
- Quantized-rateless protograph LDPC codes can be built by careful design of the protograph such that multiple puncturing patterns will still permit message passing decoding to proceed